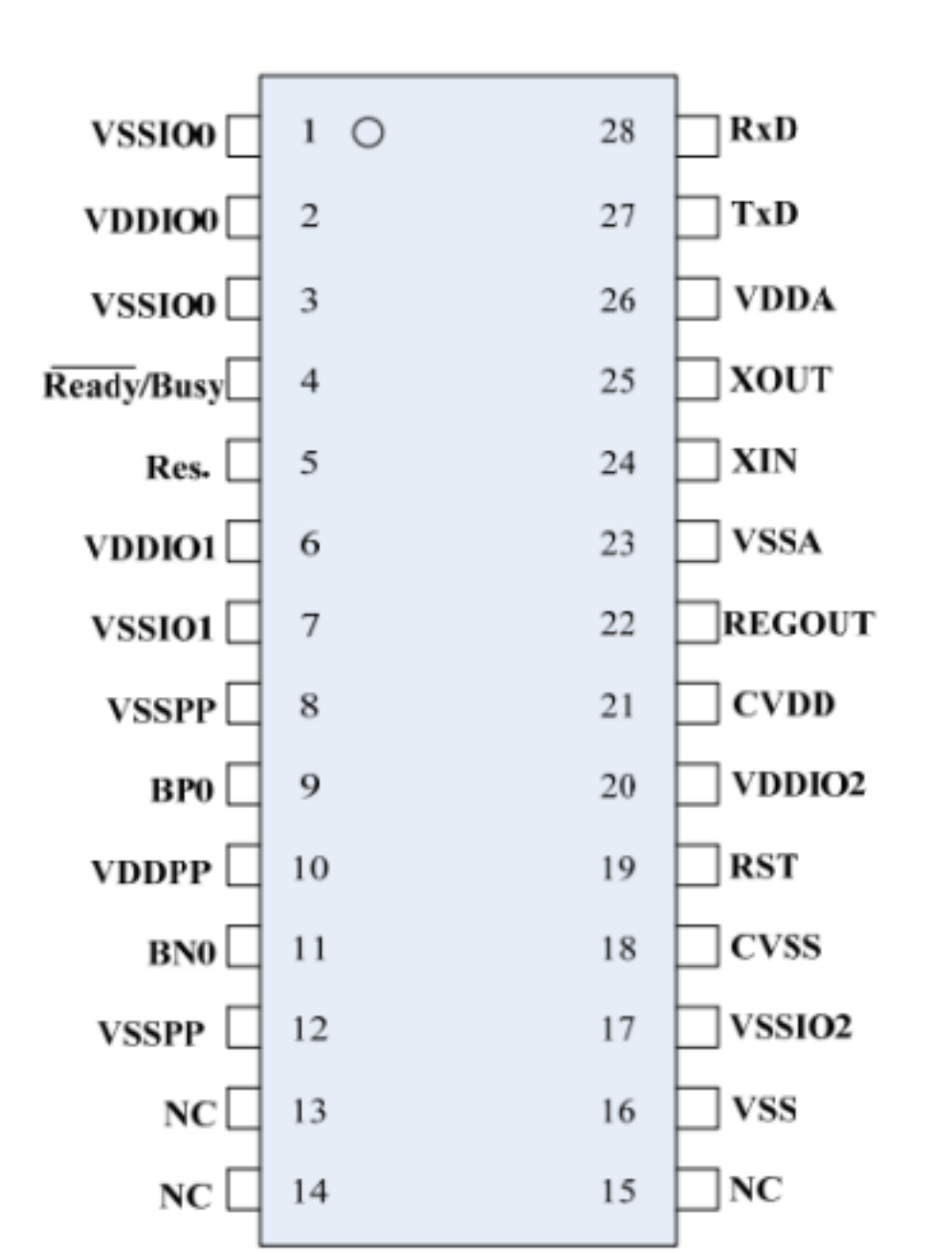


SYN6288 语音播放模块制作

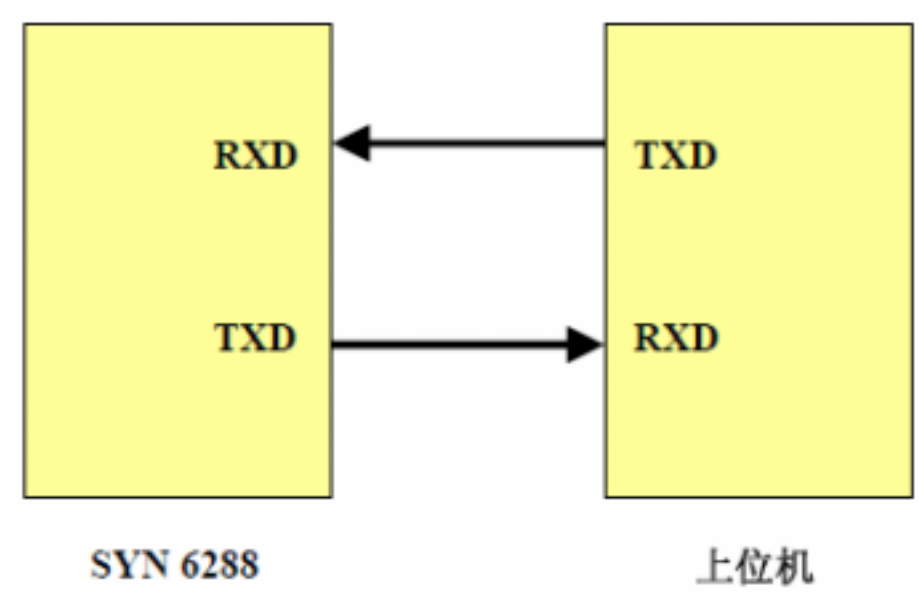
1、SYN6288 语音芯片封装图：



2、通信方式：

2.1 异步串行通讯（UART）接口

SYN6288 提供一组全双工的异步串行通讯（UART）接口，实现与微处理器或 PC 的数据传输。SYN 6288利用 TxD 和 RxD 以及 GND实现串口通信。其中 GND作为地信号。SYN 6288 芯片支持UART 接口通讯方式，



通过 UART接口接收上位机发送的命令和数据，允许发送数据的最大长度为206 字节。

## 2.2 通讯传输字节格式

- 1、初始波特率： 9600 bps
- 2、起始位：1
- 3、数据位： 8
- 4、校验位：无
- 5、停止位： 1
- 6、流控制：无

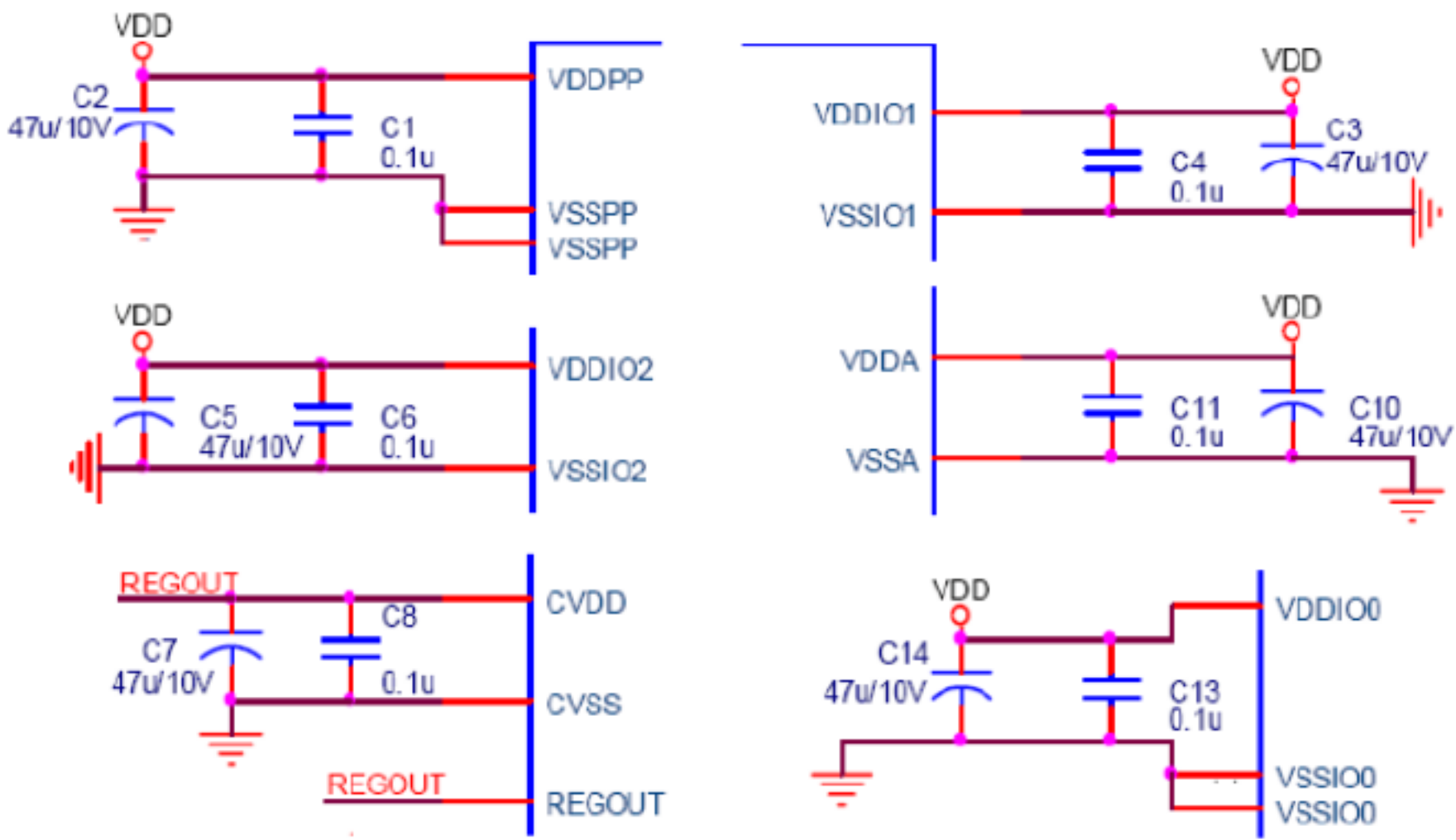
起始位	D0	D1	D2	D3	D4	D5	D6	D7	停止位
-----	----	----	----	----	----	----	----	----	-----

与 51 单片机通信时，可以用单片机的 串行通信方式 1。

## 3、硬件电路搭建：

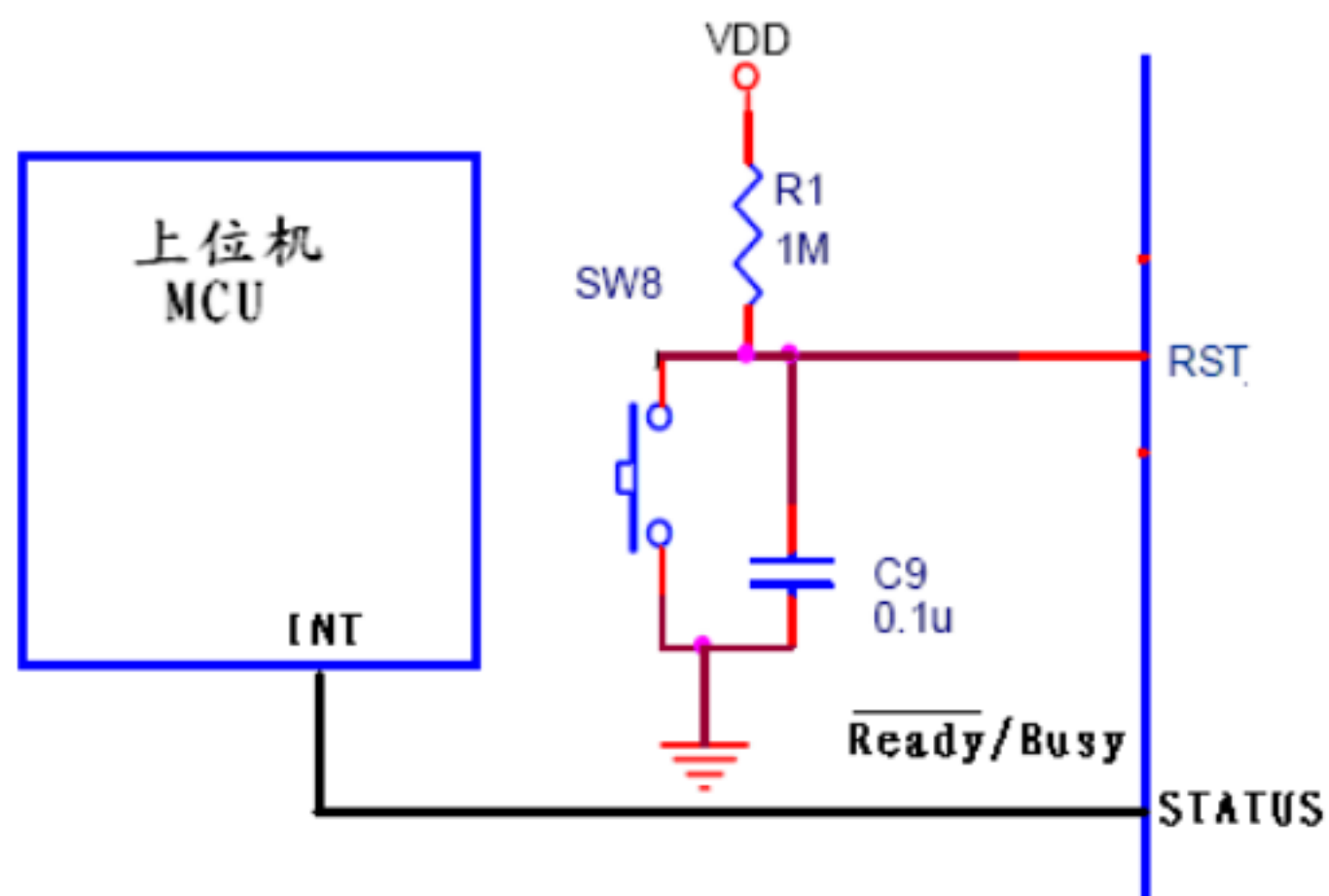
### 3.1 外接电源组接法

#### 5V 电源供电模块参考电路



备注：SYN 6288共有 6组外接电源，每组电源均使用一个 47uF和一个 0.1uF的电容；如果用户想节省成本，用户可以在每组电源上均使用 0.1uF的电容，并对 VDDPP、和 VDDA 两组电源，各加上一 47uF的电容。

### 3.2 复位电路及状态指示电路



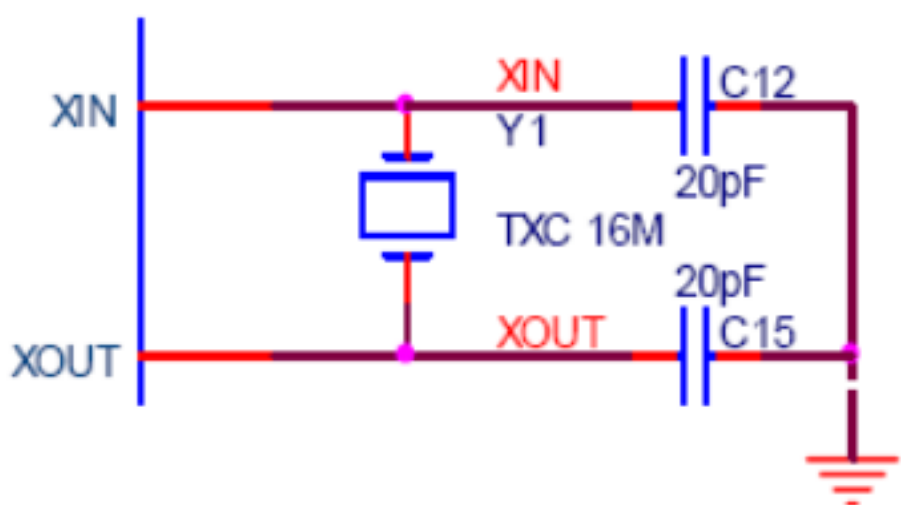
备注：Ready/Busy 此STATUS脚信号为低电平时说明芯片正在等待接收数据。在系统设计时可以将此引脚接在MCU的中断输入源上，产生一个下降沿中断请求发送数据，以示上位机MCU可以向语音合成芯片发送数据。

### 3.3 SYN6288 的扬声器输出

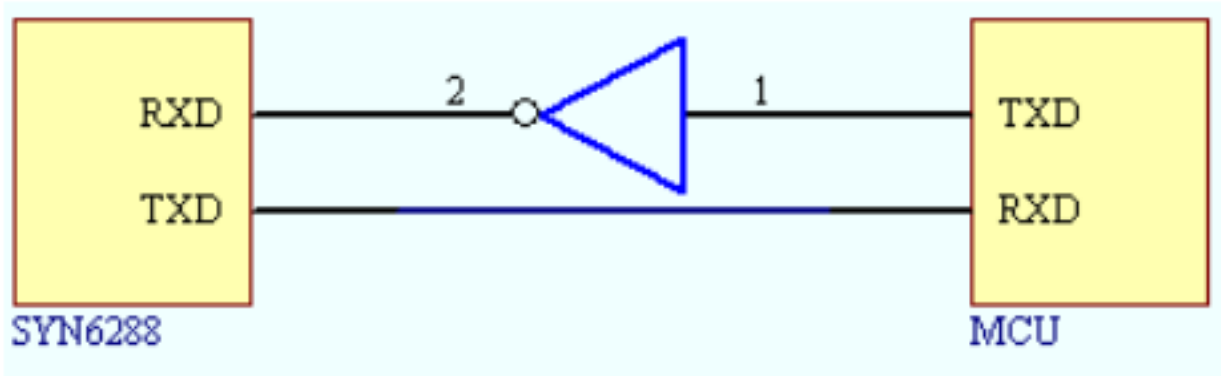
（1）为了在用户应用中输出声音，SYN6288 内置了推挽式（Push-Pull）的 DAC，可直接驱动喇叭，进行声音播报。并且 SYN6288 内置的 DAC 电路模块，使用了 VDDPP/VSSPP 供电电源模块，具体电路说明部分请参见（10.1）和（10.2）节，其供电电压值可独立于其它电源组的供电。（见右图）



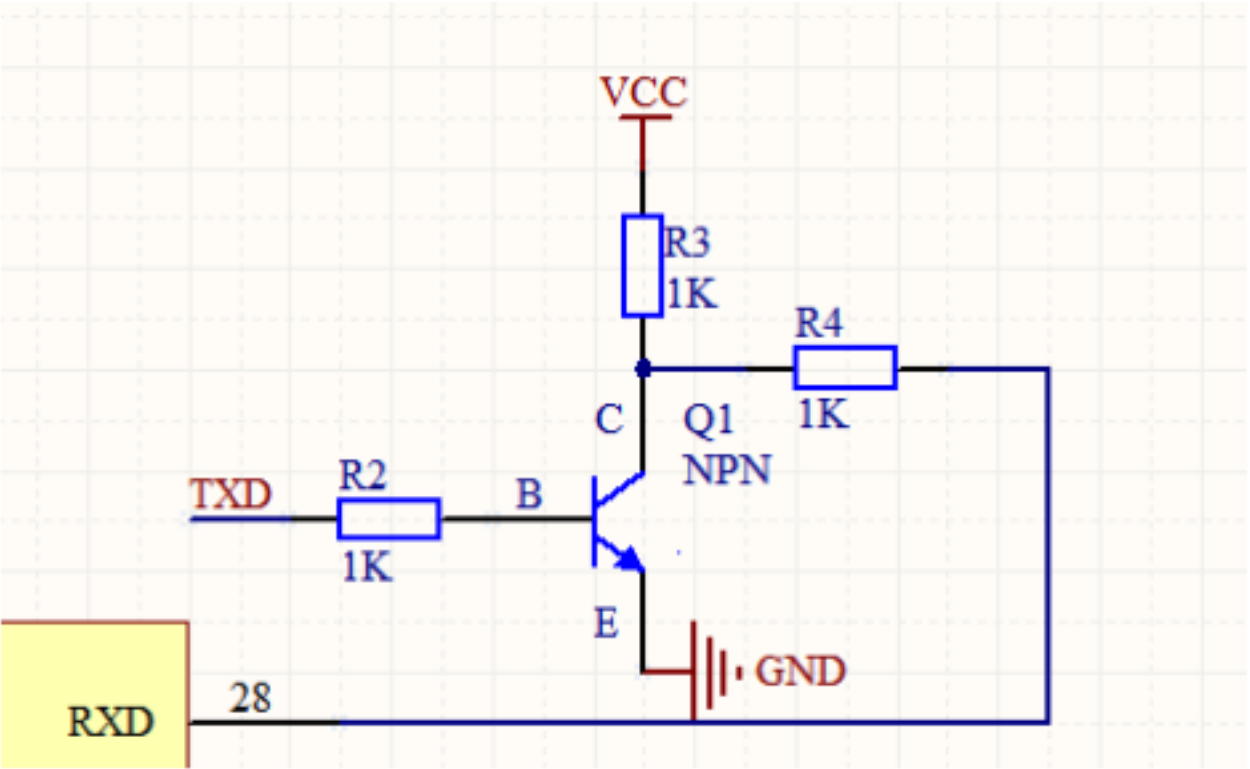
### 3.4 SYN6288 外接高速晶振



### 3.5 SYN6288 串口通信的参考电路

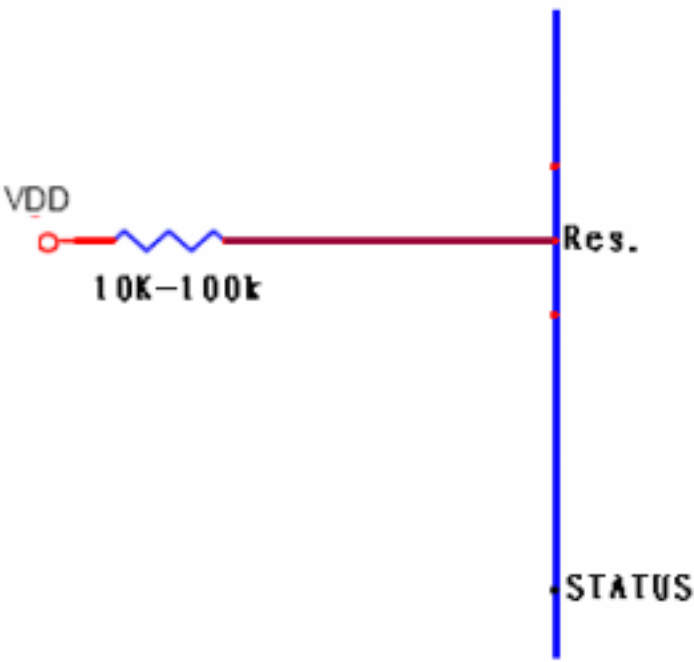


备注：上位机发送数据给 SYN6288 时，中间须加有 反向器。  
在实际电路中，我们用三极管做了一个反向器，电路如下图：

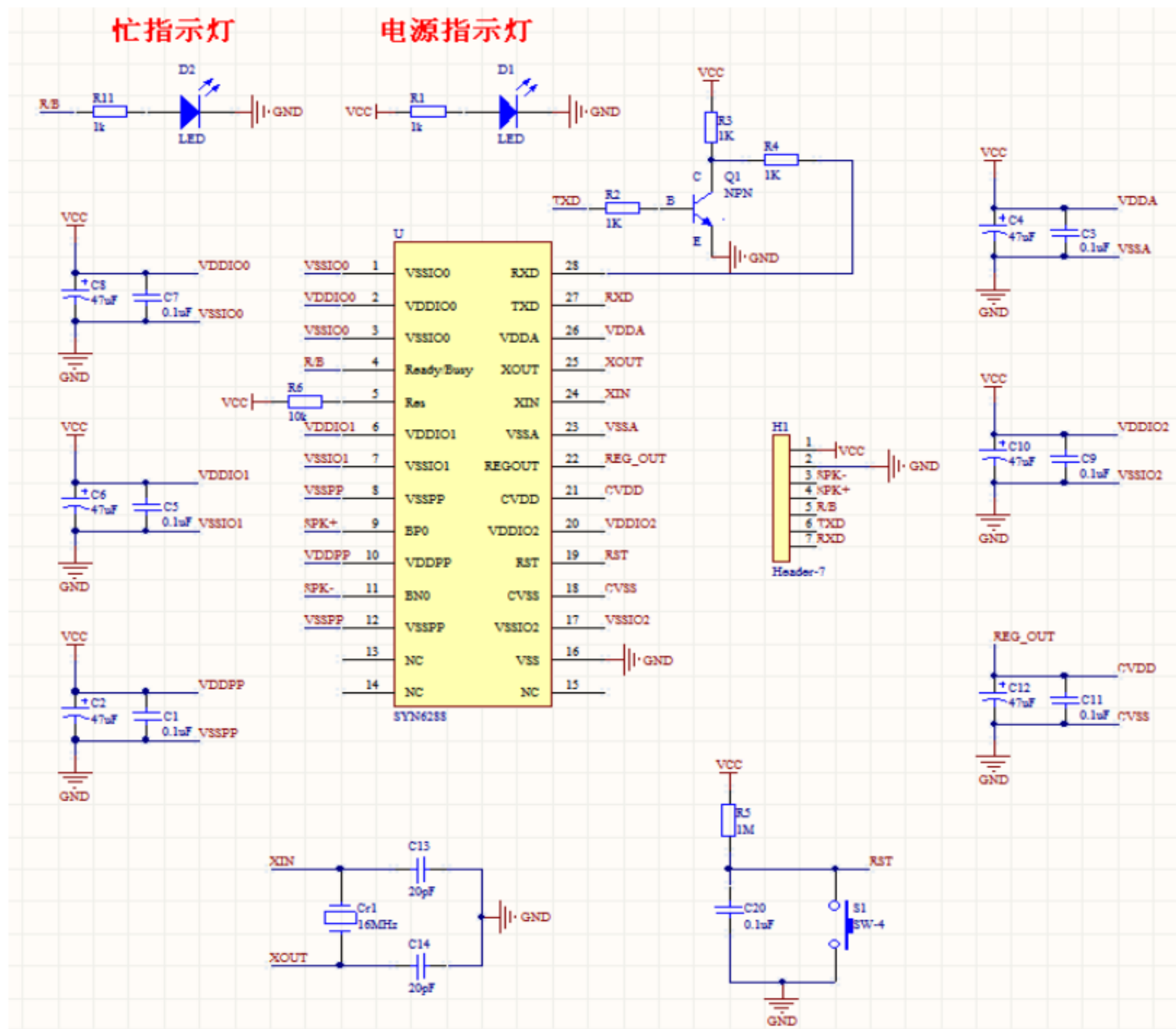


该电路的原理是： MCU 的TXD 输出电平为 0时， NPN三极管截止， RXD 收到的电平为 1。  
MCU 的TXD 输出电平为 1时， NPN三极管导通， RXD 收到的电平为 0。

### 3.6 Res 引脚（即第 5 引脚）的接法



### 3.7 总体电路原理图：



# 4、程序的编写：

## 4.1 命令帧格式：

芯片支持以下命令帧格式：“帧头 FD + 数据区长度 +数据区”格式。（最大 206个字节）

帧结构	帧头 (1字节)	数据区长度 (2字节)	数据区 (小于等于203字节)			
			命令字 (1字节)	命令参数 (1字节)	待发送文本 (小于等于200字节)	异或校验 (1字节)
数据	0xFD	0xFF 0xFF	0xFF	0xFF	0xFF . . .	0xFF
说明	定义为十六进制“0xFD”	高字节在前 低字节在后	长度必须和前面的“数据区长度”一致			

注意：数据区（含命令字，命令参数，待发送文本，异或校验）的实际长度必须与帧头后定义的数据区长度严格一致，否则芯片会报接收失败。

## 4.2 语音合成播放命令举例：

帧结构	帧头	数据区 长度	数据区			
			命令字	命令参数	待发送文本	异或校验
数据	0xFD	0x00 0x0B	0x01	0x00	字音天下 0xD3 0xEE 0xD2 0xF4 0xCC 0xEC 0xCF 0xC2	0xC1
数据帧	0xFD 0x00 0x0B 0x01 0x00 0xD3 0xEE 0xD2 0xF4 0xCC 0xEC 0xCF 0xC2 0xC1					
说明	播放文本编码格式为“GB2312”的文本“字音天下”，不带背景音乐					

上位机发送给 SYN6288 芯片的所有命令和数据都需要用“帧”的方式进行封装后传输。

其它命令请自行参考 SYN6288芯片手册：

<http://www.tts168.com.cn/SYN6288.aspx>

## 4.3 文本程序发送：

中文系统电脑的文本编码格式一般为：ANSI即GB2312 编码体系，这个可以用二进制文本编辑器（如 BinaryEditor）打开并对照 GB2312编码表察看。



## 参考程序：

### //speaker.c

```
#include <reg51.h>
#include <string.h>
#include <math.h>
#include "typedef.h"
#include "SPEAKER.h"

void delayMs(uint16 xms){
    uint16 i,j;
    for (i=0;i<xms;i++)
        for (j=0;j<123;j++);
}

//语音模块初始化
void SpeakerInit(){
    /*****          串口的初始化          *****/
    TL1=0XFA;    //在 11.0592MHZ 下，设置波特率 9600bps,工作方式 2
    TH1=0XFA;
    TMOD=0X20;
    SCON=0X50;    //串口工作方式 1，允许接收
    PCON=0X80;
    EA=0;
    REN=1;
    TI=0;        //发生中断标志位置零
    RI=0;        //接收中断标志位置零
    TR1=1;        //定时器 1用做波特率发生
}

//语音播报程序
uint8 Speaker(char * pString){
    uint8 headOfFrame[5];
    uint8 length;        //定义字符串长度
    uint8 ecc = 0;        //定义校验字节
    uint16 i = 0;

    if (pString == NULL)    //空字符串
        return -1;

    /*****          发送过程          *****/
    headOfFrame[0]=0XFD;    //构造帧头 FD
```

```

headOfFrame[1]=0X00;    //构造数据区长度的高字节
length = strlen(pString);    //需要发送文本的长度
headOfFrame[2]=length+3;// 构造数据区长度的低字节
headOfFrame[3]=0X01; // 构造命令字：合成播放命令
headOfFrame[4]=0X00; // 构造命令参数：编码格式为    GB2312

for(i=0;i<5;i++)        //依次发送构造好的 5个帧头字节
{
    ecc=ecc^(headOfFrame[i]); // 对发送的字节进行异或校验
    SBUF=headOfFrame[i];
    while (TI==0){;}        //等待发送中断标志置位
    TI=0;                    //发送中断标志位清零
}

for(i=0;i<length;i++)    //依次发送待合成的文本数据
{
    ecc=ecc^(*pString);
    SBUF = (*pString);
    pString ++;
    while(TI==0){;}
    TI=0;
}

SBUF=ecc;
while(TI==0){;}
TI=0;

return 0; //成功返回 0
}

```

//语音读整数

```

uint8 SpeakerInt(int ida){
    uint8 i;
    uint8 negative=0;    //负数标志位
    uint8 intLen=5;
    char cdat[6]={0};

    if (ida < 0){        //若为负数取绝对值
        ida = abs(ida);
        negative = 1;
    }

    cdat [0] = (char)(ida / 10000 ) ;
    cdat [1] = (char)((ida % 10000) /1000);
}

```



```

    cdat [2] = (char)((ida % 1000) /100);
    cdat [3] = (char)((ida % 100) /10);
    cdat [4] = (char)((ida % 10) /1);
    for (i=0;i<5;i++){
        cdat[i] = cdat[i] + 48;
    }

    if (cdat[0] == '0'){
        intLen = 4;
        if (cdat[1] == '0'){
            intLen = 3;
            if (cdat[2] == '0'){
                intLen = 2;
                if (cdat[3] == '0')
                    intLen = 1;
            }
        }
    }

    if (negative == 1){
        Speaker("零下");
        delayMs(1000);
    }

    Speaker(& cdat[5-intLen]);
    return 0;
}

```

//语音读小数

```

uint8 SpeakerFloat(float fda,uint8 dNum){           //fda 为传入的小数， dNum 为其小数位数
    uint8 negative=0;
    int ida;           //整数部分
    float dec;         //小数部分
    long idec;         //转化后的小数部分

    if (fda < 0){      //若为负数取绝对值
        fda = fabs(fda);
        negative = 1;
    }
    ida = floor(fda);
    dec = fda - ida;
    idec = (long) (dec * pow(10,dNum+1));           //这里多取一位数，用于处理有可能出现

```

x999的情况

```
    Speaker("[n2]");
    delayMs(200);
    if (negative == 1){
        SpeakerInt (-ida);
    }else{
        SpeakerInt (ida);
    }
    delayMs(3000);
    Speaker("点 ");
    delayMs(1000);
    Speaker("[n1]");
    delayMs(200);

    if (idec%10 >= 5){          //如果是尾数 x999的情况
        idec += 10;
    }
    idec /= 10;
    SpeakerInt(idec);
    delayMs(3000);
    Speaker("[n2]");
    delayMs(200);

    return 0;
}
```

```
void main(){
    SpeakerInit();
    while(1){

        Speaker("现在的室外温度是 ");
        delayMs(3000);
        SpeakerInt(-37);
        delayMs(1500);
        SpeakerFloat(-32.23,3);
        delayMs(3000);
        Speaker("度");
        delayMs(1000);
    }
}
```

## //Speaker.h

```
#ifndef _SPEAKER_H_
#define _SPEAKER_H_

extern void SpeakerInit();           //语音初始化程序
extern uint8 Speaker(char * pString); //语音播放程序
extern uint8 SpeakerInt(int ida);    //语音读整数

#endif
```

## //typedef.h

```
#ifndef TYPEDEF_H_
#define TYPEDEF_H_

typedef unsigned char uint8;
typedef unsigned int uint16;
typedef unsigned long uint32;

#endif
```